

**Univerzita Hradec Králové**  
Fakulta informatiky a managementu

**System pro identifikaci osob dle popisu tváře**  
3. projekt ze Znalostních technologií 2

**Petr Voborník**

UHK – FIM - im(5) – 3. ročník – ZT2 - cvičení 05

[vobornik@mikmik.cz](mailto:vobornik@mikmik.cz)

# Obsah

Obsah.....	1
Zadání .....	2
Zadání .....	2
(1) Detailní zadání: Analýza problému.....	2
(2) Detailní zadání - implementace řešení .....	3
Analýza problému .....	4
Analýza příležitosti a vhodnosti .....	4
Analýza problémové oblasti .....	4
Návrh znalostního systému.....	4
Tabulka vlastností vybraných osob .....	5
Ovládání programu .....	12
Popis zdrojového kódu .....	13
Fakta .....	13
Načtení možností jednotlivých charakteristik .....	13
Stanovení pořadí .....	14
Načtení vlastností .....	14
Výjimky.....	14
Vyhledání výsledku .....	14
Odstranění či obnovení pomocných faktů .....	14
Dotaz na pokračování .....	14
Závěr .....	15
Zdrojový kód .....	16
Program (Tvare.clp) .....	16
Data (data.clp) .....	26
Help (help.clp).....	27

# Zadání

V rámci třetího samostatného bodovaného úkolu budete pracovat na vytvoření vlastního znalostního systému.

## Zadání

Úkol má dvě části: nejprve zpracujete [\(1\) analýzu](#) a potom se pustíte do psaní [\(2\) programu](#). Doména, ve které budete pracovat, vám byla/bude schválena na cvičení. Nedílnou součástí tohoto úkolu je též úspěšné předvedení výsledného znalostního systému na cvičení.

### (1) Detailní zadání: Analýza problému

Analýza má alespoň zjednodušeně pokrýt některé etapy znalostního projektu (viz přednášky). Jejím výsledkem bude dokument, v němž zaznamenate všechny své úvahy. Měli byste se dotknout zejména těchto bodů:

#### a) Analýza příležitosti a vhodnosti

- úvodní popis problémové domény a typických úloh, které se v ní řeší

*Příklad: vysvětlení, že Váš systém umožní rozpoznávat nemoci papoušků + příklad, kdy je takové rozpoznání obtížné a proč...*

- důvody, proč by bylo vhodné vytvořit ZS, jeho výhody ve srovnání s jinými řešeními

*Příklad: rostoucí počty papoušků v českých domácnostech, nedostatek odborníků, kteří by měli čas dělit se o zkušenosti s amatérskými chovateli...*

- upozornění na možná rizika

*Příklad: chybějící česká terminologie pro popis příznaků exotických chorob bude komplikací při tvorbě amatérům srozumitelného rozhraní...*

#### b) Analýza problémové oblasti

- základní pojmy a jejich vztahy zvolené části problémové oblasti

*Příklad: konkrétní nemoci papoušků, příznaky těchto nemocí...*

#### c) Návrh znalostního systému

- vymezení, co bude a co nebude systém řešit

*Příklad: pouze papoušci z Austrálie...*

- popis báze znalostí: rozhodovací strom nebo rozhodovací tabulky, z které bude zřejmé, s jakými fakty budete pracovat a jak je budete využívat. Pokud pracujete s materiály v tištěné podobě, volte vhodné zjednodušení.

*Příklad: viz tabulka [papousek.xls](#)*

- zdroje a metody získání znalostí pro ZS

*Příklad: rozhovor s chovatelem, zápis vlastních zkušeností, literatura...*

### **Než se pustíte do práce na analýze ...**

1. Při psaní analýzy neuvažujte o pozdější implementaci, tj. nezjednodušujte si zadání jen proto, že třeba neumíte dobře psát programy v CLIPSu. Vytvořte ji, jak nejlépe budete umět.
2. Jako vodítko pro stanovení přiměřeného rozsahu zpracované problémové oblasti uvažujte o 25-30 objektech. Na objektech by měly být definovány 4 až 6 vlastností. Definiční obor každé vlastnosti by měl pozůstat ze 4 až 6 hodnot.
3. Výše uvedené příklady jsou jen ilustrační, vaše popisy by měly být detailnější.

## **(2) Detailní zadání - implementace řešení**

Podle analýzy zvolené problémové oblasti vytvořte implementaci v CLIPSu.

### **Výsledný program by měl umět:**

1. K zachycení vlastností známých objektů (dle vypracované analýzy) bude vytvořena vhodná šablona.
2. Známé objekty budou vloženy do báze faktů (interní nebo externí).
3. Kontrolujte údaje načítané od uživatele, jestli vyhovují povoleným hodnotám příslušné vlastnosti.
4. Požadavky uživatele na zkoumaný objekt vložte do báze faktů.
5. Uživatel bude mít možnost zadat odpověď "nevím" místo konkrétní odpovědi.
6. Program nalezne objekt, který bude splňovat všechny požadavky uživatele. Bude ošetřen i případ, kdy takový objekt neexistuje v bázi známých objektů. Jestli vyhovuje víc objektů, budou nalezeny všechny.
7. Výsledek odvozování bude vždy vložen jako fakt do báze faktů a poté bude vypsán.
8. Po identifikaci objektu bude uživatel tázán, jestli si přeje pokračovat identifikací dalšího objektu.

# **Analýza problému**

## ***Analýza příležitosti a vhodnosti***

Program umožňuje identifikaci osob na základě slovního popisu vzhledu tváře hledané osoby. Tato identifikace není vždy snadná, neboť i na první pohled odlišné osoby mohou mít shodné rysy, které takovému systému neumožní je od sebe odlišit. Například Reye Korantenga a Bila Cosbyho – dvě úplně jiné osoby i různé ve tváři – v systému rozlišují pouze Reyovy kotlety, které Bil nenosí.






## ***Analýza problémové oblasti***






Tento znalostní systém může pomoci policii k rychlé identifikaci hledané osoby na základě výpovědi očitého svědka. Několika málo jednoduchými otázkami se lze dopátrat hledané osoby, či značně zúžit jejich okruh, a na základě fotografií těchto několika osob velmi rychle a přesně identifikovat a nalézt hledanou osobu.

## ***Návrh znalostního systému***

Pro definování podstatných vzhledových vlastností tváří osob, bylo náhodně vybráno 37 známých osobností (aby si již dle jména každý mohl tvář osoby živě představit), na jejichž tvářích bylo určeno 6 + 2 další vlastnosti, na jejichž základě se dá každá jednoznačně identifikovat. Existence vlastnosti „další“ dává možnost i do budoucna přidávat nové vlastnosti (pihy, vrásky, mateřská znaménka...), které i na dále mohou nově přidaným osobám zajistit jedinečnost v databázi.






**Tabulka vlastností vybraných osob**

Fotka	Příjmení	Jméno	Pohlaví	Rasa	Vlasy	Barva vlasů	Barva očí	Vousy	Další
	Alda	Alan	muž	běloch	pěšinka	černá	černá	žádné	
	Allen	Woody	muž	běloch	pěšinka	šedá	hnědá	žádné	brýle
	Applegate	Christina	žena	běloch	pod uši	blond	modrá	-	
	Basiková	Bára	žena	běloch	pod uši	zrzavá	modrá	-	
	Bílá	Lucie	žena	běloch	po ramena	černá	hnědá	-	







	Collins	Joan	žena	běloch	kštice	černá	černá	-	
	Connery	Sean	muž	běloch	pleš	-	hnědá	všude	kotlety
	Cosby	Bill	muž	černoch	kudrlinky	černá	černá	žádné	
	Falk	Peter	muž	běloch	pěšinka	prošedivělá	černá	žádné	
	Fišer	Bohumil	muž	běloch	pěšinka	hnědá	hnědá	žádné	brýle

	Flair	Ric	muž	běloch	pěšinka	blond	modrá	žádné	
	Ford	Harrison	muž	běloch	dlouhý ježek	prošedivělá	modrá	žádné	
	Frakes	Jonathan	muž	běloch	dlouhý ježek	černá	modrá	všude	kotlety
	Freeman	Morgan	muž	černoch	dlouhý ježek	prošedivělá	černá	okolo	
	Funes	Louis De	muž	běloch	pleš	-	černá	žádné	kotlety



	Goldberg	Whoopi	žena	černoch	po ramena	černá	černá	-	
	Gott	Karel	muž	běloch	pěšinka	černá	černá	žádné	kotlety
	Hogan	Hulk	muž	běloch	okolo hlavy	blond	modrá	knír	kotlety
	Houston	Whitney	žena	černoch	pod uši	hnědá	hnědá	-	
	Chan	Jackie	muž	asiat	pěšinka	černá	černá	žádné	

	Klaus	Václav	muž	běloch	ježek	šedá	černá	knír	brýle
	Koranteng	Rey	muž	černoch	kudrlinky	černá	černá	žádné	kotlety
	Lamas	Lorenzo	muž	běloch	po ramena	černá	hnědá	žádné	kotlety
	Li	Jet	muž	asiat	dlouhý ježek	černá	hnědá	žádné	
	Novotný	Petr	muž	běloch	pěšinka	černá	hnědá	okolo	

	O'Neill	Ed	muž	běloch	dlouhý ježek	černá	hnědá	žádné	
	Ruml	Jan	muž	běloch	pleš	-	modrá	všude	kotlety
	Sagal	Katey	žena	běloch	po ramena	zrzavá	černá	-	
	Savalas	Telly	muž	běloch	pleš	-	černá	žádné	
	Schwarzenegger	Arnold	muž	běloch	dlouhý ježek	černá	modrá	žádné	
	Smith	Will	muž	černoch	ježek	černá	černá	okolo	

	Součková	Marie	žena	běloch	kštice	zrzavá	modrá	-	kotlety
	Stewart	Patrick	muž	běloch	pleš	-	modrá	žádné	
	Špidla	Vladimír	muž	běloch	pěšinka	šedá	černá	žádné	brýle
	T	Booker	muž	černoch	ježek	černá	černá	knír	
	Zázvorková	Stella	žena	běloch	po ramena	šedá	černá	-	
	Zeman	Miloš	muž	běloch	dlouhý ježek	šedá	modrá	žádné	

## Ovládání programu

- Uživatel spustí program (načte jej – File Open, Load Buffer; resetuje Ctrl+E a spustí Ctrl+R).
- Uživateli jsou vypsána pravidla ovládání programu, například, že může zadávání kdykoli ukončit zadáním slova „konec“, nebo pokud neví, která z možných hodnot je ta pravá, může zadat „nevím“, případně se na ni dotázat příkazem „help“ + neznámá hodnota.
- Dále je uživatel okamžitě dotázán na první vzhledovou vlastnost tváře hledané osoby. Která vlastnost to bude závisí na tom, která ze všech možných, dosud nedotázaných, má největší vypovídající schopnost (nejvíce možných hodnot), což je zjištěno ze všech tváří uložených v databázi.
- Při zadávání hodnot platí že lze zadat pouze hodnotu, která je zde uvedena ve výčtu, nebo „nevím“ (není-li tato hodnota známa nebo ji výčet možností nenabízí), „help“ + název nabízené hodnoty (pro bližší popis této hodnoty), případně „konec“, který doplní všechny dosud nezadané vlastnosti hodnotou „nevím“. Vždy se zadává pouze jedna z hodnot, jedinou výjimkou jsou „další vlastnosti“, kterých může být i více. Ty je třeba vypsát všechny oddělené mezerou. Vypsání hodnot(y) je třeba potvrdit stiskem klávesy Enter.
- Takto je uživatel dotazován, dokud nejdou buď zadány všechny možné popisované vlastnosti tváře, nebo pokud z již zadaných vlastností není jednoznačně určena jediná (případně žádná) osoba. V tom případě je tato skutečnost uživateli oznámena a vyhledávání ukončeno. Pokud by uživatel měl k zadání vlastnosti na výběr pouze jedinou možnost, program ji zadá za něj.
- Výběr možností je v každém kroku znovu upraven pouze z výčtu osob, které již zadaným vlastnostem odpovídají.
- Na konec je vypsán výsledek hledání. Může to být buď jeden či více osob, které odpovídají zadaným vlastnostem nebo oznámení, že osoba s těmito vzhledovými vlastnostmi tváře se v databázi nevyskytuje.
- Dále je uživatel dotázán, přeje-li si pokračovat další identifikací (začít od začátku). Zadá-li jedničku (1), program se spustí znovu, zadá-li nulu (0), program se ukončí.

## Popis zdrojového kódu

Program je vytvořen tak, aby uživateli maximálně usnadnil práci s ním. Jde například o možnost vypsání nápovědy ke všem zde používaným výrazům. Na výběr jsou dávány pokaždé pouze ty hodnoty, které jsou možné – pokaždé jsou znovu vybrány z databáze osob, avšak vždy již pouze těch, které vyhovují již zadaným vlastnostem. Pořadí dotazování se na vlastnosti je takové aby co nejdříve vedlo k jednoznačnému výsledku – identifikaci jediné hledané osoby. Vyhovuje-li jich zadaným vlastnostem více, jsou nakonec vypsány všechny. Program také za uživatele sám doplňuje vlastnosti, které by již bylo možné klasifikovat pouze jedinou hodnotou, čímž celý proces urychluje a zjednodušuje. Možnost zadání hodnoty „nevím“ uživateli dovoluje nezadat hodnotu vlastnosti, kterou si není zcela jist. Identifikaci je také možno ukončit zadáním hodnoty „konec“, která zbývajícím vlastnostem automaticky přiřadí hodnotu „nevím“.

A nyní blíže k jednotlivým částem programu.

### **Fakta**

Oddíl fakta obsahuje definici šablony „**tvare**“ a definici potřebných fakt. Fakta „**moznosti**“ slouží jako seznam, který si program teprve naplní možnými hodnotami jednotlivých vlastností. Fakta „**vstupy**“ program zase v budoucnu použije pro uložení hodnot jednotlivých vlastností zadaných uživatelem. Zatím je u všech „nevím“, což umožní se na ně odkazovat i před tím, než je uživatel zadá. To, které vlastnosti má uživatel ještě zadat, je uloženo ve faktu „**poradi**“, které se na počátku zkopíruje z faktu „zaloha-poradi“ (ve faktu „pokusne-poradi“ probíhá seřazení tohoto seznamu dle vypovídací hodnoty každé z vlastností v seznamu). Fakta „**zadej**“ obsahují texty, které se vypíší jako výzva uživateli pro zadání hodnoty určité vlastnosti (aby se toto dalo řešit jediným univerzálním pravidlem). Seznam „**povolene\_first**“ určuje, které další hodnoty mimo povolených ve výčtu možností smí uživatel zadat, aby program pokračoval dále. Fakt „**start**“ pak definuje aktuální fázi programu – tedy že má program začít.

Databáze znalostí a nápovědné texty jsou načítány z externích souborů (data.clp a help.clp). O jejich načtení se stará sem právem zařazené pravidlo „nacti-data“ (je spouštěno pouze 1x i při opakované identifikaci). Pravidlo „start“ pak vypíše úvodní informace o programu (je spouštěno vždy před začátkem identifikace).

### **Načtení možností jednotlivých charakteristik**

Jednotlivá pravidla v tomto oddílu doplní seznamy možných hodnot jednotlivých vlastností s ohledem na vlastnosti již zadané (bere v úvahu hodnoty vlastností pouze těch osob, které filtrem již zadaných vlastností prošly). Pravidlo „doplň-vlastnost-je-li-jen-jedna“ automaticky doplní hodnotu vstupu příslušné

vlastnosti v případě, že existuje již jen jediná možná hodnota, která lze dané vlastnosti přiřadit (po profiltrování osob na základě již zadaných vlastností).

### ***Stanovení pořadí***

Přeuspořádá pořadí vlastností tak, aby první z nich měla největší vypovídající hodnotu (aby existoval největší možný počet hodnot, které vlastnosti lze přiřadit). Na závěr je toto pořadí uloženo do faktu „poradí“, podle nějž program bude dále pokračovat v dotazování.

### ***Načtení vlastností***

Zde je pravidlo „nacitej-dalsi-vlastnost“, které od uživatele načte hodnotu vlastnosti, která je právě na řadě. Toto pravidlo slouží k zadávání všech vlastností (je univerzální). Další pravidla se pak starají o zpracování zadané hodnoty. Ta se buď uloží jako vstup, případně (jedná-li se o speciální hodnotu typu příkaz) je tato zpracována. Jde především o vypsání nápovědy, ukončení zadávání (okamžité vyhledání možných osob) a ukončení v případě, že již bylo zadány všechny vlastnosti.

### ***Výjimky***

Zde je ošetřena možnost, aby se program nedotazoval uživatele na barvu vlasů hledané osoby, když už před tím zadal, že osoba je plešatá.

### ***Vyhledání výsledku***

Tato pravidla najdou a vypíší jména všech osob, které vyhovují zadaným vlastnostem. Pokud taková v databázi není, je o této skutečnosti uživatel informován.

### ***Odstranění či obnovení pomocných faktů***

Pravidla obnoví výchozí nastavení faktů, aby se případně mohlo s identifikací začít od začátku a v bázi faktů nezůstal nepořádek.

### ***Dotaz na pokračování***

Program se uživatele zeptá, přeje-li si začít novou identifikaci osoby (1), či program ukončit (0). Zadanou volbu ihned vykoná.

## **Závěr**

Zvolený problém se podařilo úspěšně vyřešit programem, který je uživatelsky velmi přívětivý a svou univerzálností umožňuje své použití i do budoucna, po rozšíření databáze osob a jejich vlastností.



# Zdrojový kód

## Program (Tvare.clp)

```
; Program na identifikace osob podle vzhledu jejich tvari
;
; © 2004 Petr Vobornik

; ----- Fakta -----

(deftemplate tvar
  (slot prijmeni)
  (slot jmeno)
  (slot pohlavi)
  (slot rasa)
  (slot vlasy)
  (slot vlasy_b)
  (slot oci)
  (slot vousy (default zadne))
  (multislot dalsi)
)

(deffacts data

  (moznosti pohlavi)
  (moznosti rasa)
  (moznosti vlasy)
  (moznosti vlasy_b)
  (moznosti oci)
  (moznosti vousy)
  (moznosti dalsi)

  (vstup pohlavi nevim)
  (vstup rasa     nevim)
  (vstup vlasy   nevim)
  (vstup vlasy_b nevim)
  (vstup oci     nevim)
  (vstup vousy   nevim)
  (vstup dalsi   nevim)

  (zaloha-poradi pohlavi rasa vlasy vlasy_b oci vousy dalsi)
  (pokusne-poradi pohlavi rasa vlasy vlasy_b oci vousy dalsi)

  (zadej pohlavi "Zadejte pohlavi hledane osoby.")
  (zadej oci     "Zadejte barvu oci hledane osoby.")
  (zadej vlasy  "Zadejte druh ucesu hledane osoby.")
  (zadej rasa   "Zadejte rasu hledane osoby.")
  (zadej vousy  "Zadejte typ vousu hledane osoby.")
  (zadej dalsi  "Zadejte, které dalsi rysy ma hledana osoba. Je-li jich
vice, vypiste je vsechny a oddelte je mezerou.")
  (zadej vlasy_b "Zadejte barvu vlasu hledane osoby.")

  (xx xx)
  (povolene_first help konec nevim)
  (start)
)

(defrule nacti-data
  (declare (salience 10))
```

```

(start)
(not(tvar))
=>
(printout t "Nacitam data..." crlf crlf)
(load-facts "data.clp")
(load-facts "help.clp")
)

(defrule start
  ?fakt <- (start)
  =>
  (printout t "Vitejte v programu na identifikaci osob podle popisu tvare."
crlf crlf
          "Zadavejte presne pozadovane hodnoty a volbu vzdy potvrďte
enterem." crlf
          "Nebudete-li si jisti, vzdy muzete zadat hodnotu 'nevim'."
crlf
          "Jste-li si vsak hodnotu jisti, ale ta neni mezi nabizenymi,
pak hledana osoba nejspis neni evidovana v databazi." crlf
          "Pokud nebudete rozumet nektere z nabizenych hodnot, staci
zadat 'help' a pripade nazev teto hodnoty." crlf
          "Zadavani hodnot muzete kdykoli ukoncit zadanim prikazu
'konec' a vypisi se osoby odpovidajici do te doby zadany vlastnostem."
crlf crlf)
  (retract ?fakt)
  (assert (priprav))
)

; ----- Nacteni moznosti jednotlivych charakteristik -----

(defrule dopln-pohlavi
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ pohlavi $?poradi))
  ?f <- (moznosti pohlavi $?hodnoty)
  (vstup rasa      ?vrasa)
  (vstup vlasy     ?vvlasy)
  (vstup vlasy_b   ?vvlasy_b)
  (vstup oci       ?voci)
  (vstup vousy     ?vvousy)
  (vstup dalsi     $?vdalsi)
  (tvar (pohlavi ?pohlavi)
        (rasa      ?rasa&:( or (eq ?rasa      ?vrasa      ) (eq ?vrasa      nevim)
        ))
        (vlasy     ?vlasy&:( or (eq ?vlasy     ?vvlasy     ) (eq ?vvlasy     nevim)
        ))
        (vlasy_b   ?vlasy_b&:(or (eq ?vlasy_b   ?vvlasy_b) (eq ?vvlasy_b nevim)
        ))
        (oci       ?oci&:( or (eq ?oci       ?voci       ) (eq ?voci       nevim)
        ))
        (vousy     ?vousy&:( or (eq ?vousy     ?vvousy     ) (eq ?vvousy     nevim)
        ))
        (dalsi     $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
        $?vdalsi)))
  )
  (test (not (member$ ?pohlavi $?hodnoty)))
  =>
  (retract ?f)
  (assert (moznosti pohlavi $?hodnoty ?pohlavi))
)

```

```

(defrule dopln-rasa
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ rasa $?poradi))
  ?f <- (moznosti rasa $?hodnoty)
  (vstup pohlavi ?vpohlavi)
  (vstup vlasy ?vvlasy)
  (vstup vlasy_b ?vvlasy_b)
  (vstup oci ?voci)
  (vstup vousy ?vvousy)
  (vstup dalsi $?vdalsi)
  (tvar (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa ?rasa)
    (vlasy ?vlasy&:( or (eq ?vlasy ?vvlasy ) (eq ?vvlasy nevim)
))
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
))
    (oci ?oci&:( or (eq ?oci ?voci ) (eq ?voci nevim)
))
    (vousy ?vousy&:( or (eq ?vousy ?vvousy ) (eq ?vvousy nevim)
))
    (dalsi $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
  $?vdalsi)))
  )
  (test (not (member$ ?rasa $?hodnoty)))
  =>
  (retract ?f)
  (assert (moznosti rasa $?hodnoty ?rasa))
)

```

```

(defrule dopln-vlasy
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ vlasy $?poradi))
  ?f <- (moznosti vlasy $?hodnoty)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa ?vrasa)
  (vstup vlasy_b ?vvlasy_b)
  (vstup oci ?voci)
  (vstup vousy ?vvousy)
  (vstup dalsi $?vdalsi)
  (tvar (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa ?rasa&:( or (eq ?rasa ?vrasa ) (eq ?vrasa nevim)
))
    (vlasy ?vlasy)
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
))
    (oci ?oci&:( or (eq ?oci ?voci ) (eq ?voci nevim)
))
    (vousy ?vousy&:( or (eq ?vousy ?vvousy ) (eq ?vvousy nevim)
))
    (dalsi $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
  $?vdalsi)))
  )
  (test (not (member$ ?vlasy $?hodnoty)))
  =>
  (retract ?f)
  (assert (moznosti vlasy $?hodnoty ?vlasy))
)

```

```

(defrule dopln-vlasy_b
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ vlasy_b $?poradi))
  ?f <- (moznosti vlasy_b $?hodnoty)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa ?vrasa)
  (vstup vlasy ?vvlasy)
  (vstup oci ?voci)
  (vstup vousy ?vvousy)
  (vstup dalsi $?vdalsi)
  (tvar (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa ?rasa&:( or (eq ?rasa ?vrasa ) (eq ?vrasa nevim)
))
    (vlasy ?vlasy&:( or (eq ?vlasy ?vvlasy ) (eq ?vvlasy nevim)
))
    (vlasy_b ?vlasy_b)
    (oci ?oci&:( or (eq ?oci ?voci ) (eq ?voci nevim)
))
    (vousy ?vousy&:( or (eq ?vousy ?vvousy ) (eq ?vvousy nevim)
))
    (dalsi $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
  $?vdalsi)))
  )
  (test (not (member$ ?vlasy_b $?hodnoty)))
  =>
  (retract ?f)
  (assert (moznosti vlasy_b $?hodnoty ?vlasy_b))
)

(defrule dopln-oci
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ oci $?poradi))
  ?f <- (moznosti oci $?hodnoty)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa ?vrasa)
  (vstup vlasy ?vvlasy)
  (vstup vlasy_b ?vvlasy_b)
  (vstup vousy ?vvousy)
  (vstup dalsi $?vdalsi)
  (tvar (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa ?rasa&:( or (eq ?rasa ?vrasa ) (eq ?vrasa nevim)
))
    (vlasy ?vlasy&:( or (eq ?vlasy ?vvlasy ) (eq ?vvlasy nevim)
))
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
))
    (oci ?oci)
    (vousy ?vousy&:( or (eq ?vousy ?vvousy ) (eq ?vvousy nevim)
))
    (dalsi $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
  $?vdalsi)))
  )
  (test (not (member$ ?oci $?hodnoty)))
  =>
  (retract ?f)
  (assert (moznosti oci $?hodnoty ?oci))
)

```

```

)
(defrule dopln-vousy
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ vousy $?poradi))
  ?f <- (moznosti vousy $?hodnoty)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa ?vrasa)
  (vstup vlasy ?vvlasy)
  (vstup vlasy_b ?vvlasy_b)
  (vstup oci ?voci)
  (vstup dalsi $?vdalsi)
  (tvar (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa ?rasa&:( or (eq ?rasa ?vrasa ) (eq ?vrasa nevim)
))
    (vlasy ?vlasy&:( or (eq ?vlasy ?vvlasy ) (eq ?vvlasy nevim)
))
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
))
    (oci ?oci&:( or (eq ?oci ?voci ) (eq ?voci nevim)
))
    (vousy ?vousy)
    (dalsi $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
$?vdalsi)))
  )
  (test (not (member$ ?vousy $?hodnoty)))
  =>
  (retract ?f)
  (assert (moznosti vousy $?hodnoty ?vousy))
)

(defrule dopln-dalsi
  (priprav)
  (pokusne-poradi $?poradi)
  (test (member$ dalsi $?poradi))
  ?f <- (moznosti dalsi $?hodnoty)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa ?vrasa)
  (vstup vlasy ?vvlasy)
  (vstup vlasy_b ?vvlasy_b)
  (vstup oci ?voci)
  (vstup vousy ?vvousy)
  (tvar (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa ?rasa&:( or (eq ?rasa ?vrasa ) (eq ?vrasa nevim)
))
    (vlasy ?vlasy&:( or (eq ?vlasy ?vvlasy ) (eq ?vvlasy nevim)
))
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
))
    (oci ?oci&:( or (eq ?oci ?voci ) (eq ?voci nevim)
))
    (vousy ?vousy&:( or (eq ?vousy ?vvousy ) (eq ?vvousy nevim)
))
    (dalsi $? ?dalsi $?)
  )
  (test (not (member$ ?dalsi $?hodnoty)))
  =>
  (retract ?f)

```

```

    (assert (moznosti dalsi $?hodnoty ?dalsi))
  )

(defrule dopln-vlastnost-je-li-jen-jedna
  (declare (salience -5))
  (priprav)
  (moznosti ?vlastnost ?moznost)
  ?fakt <- (pokusne-poradi $?pred ?vlastnost $?za)
  ?fakt2 <- (vstup ?vlastnost nevim)
  =>
  (retract ?fakt ?fakt2)
  (assert (vstup ?vlastnost ?moznost))
  (assert (pokusne-poradi $?pred $?za))
)

; ----- Stanoveni poradi -----

(defrule usporadej-pokusne-poradi
  (declare (salience -10))
  (priprav)
  ?fakt <- (pokusne-poradi $?zacatek ?vlastnost1 $?stred ?vlastnost2
  $?konec)
  (moznosti ?vlastnost1 $?seznam-vlastnosti1)
  (moznosti ?vlastnost2 $?seznam-vlastnosti2)
  (test (< (length$ $?seznam-vlastnosti1)
           (length$ $?seznam-vlastnosti2)
          ))
  =>
  (retract ?fakt)
  (assert (pokusne-poradi $?zacatek ?vlastnost2 $?stred ?vlastnost1
  $?konec))
)

(defrule vytvor-konecne-poradi
  (declare (salience -15))
  ?faze <- (priprav)
  ?fakt <- (pokusne-poradi $?poradi)
  =>
  (retract ?fakt ?faze)
  (assert (poradi $?poradi))
  (assert (nacitej))
)

; ----- Nacteni vlastnosti -----

(defrule nacitej-dalsi-vlastnost
  (nacitej)
  (not(hodnota $?hodnota))
  (poradi ?vlastnost $?ostatni)
  (moznosti ?vlastnost $?moznosti)
  (zadej ?vlastnost ?text)
  (povolene_first $?povolene_first)
  (xx $?xx)
  =>
  (printout t ?text crlf
            "Povolene hodnoty jsou: " $?moznosti " a nevim." crlf)
  (bind $?hodnoty $?xx)
  (while (or (and (not(member$ $?hodnoty $?moznosti))
                 (not(member$ (first$ $?hodnoty) $?povolene_first)))

```

```

        (eq 0 (length$ $?hodnoty))) do
      (bind $?hodnoty (explode$ (readline)))
    )
    (assert (hodnota ?vlastnost $?hodnoty))
    (printout t crlf)
  )

(defrule zpracuj-zadany-konec
  (declare (salience 30))
  ?fakt <- (nacitej)
  ?fakt2 <- (hodnota ?vlastnost konec)
  =>
  (retract ?fakt ?fakt2)
  (assert (hledej))
)

(defrule zpracuj-zadany-help-existuje
  (declare (salience 20))
  (nacitej)
  ?fakt <- (hodnota ?vlastnost help $?nevi)
  (help ?vlastnost $?nevi ?text)
  =>
  (printout t $?text crlf crlf)
  (retract ?fakt)
)

(defrule zpracuj-zadany-help-neexistuje
  (declare (salience 20))
  (nacitej)
  ?fakt <- (hodnota ?vlastnost help $?nevi)
  (not(help ?vlastnost $?nevi ?text))
  =>
  (printout t "K zadane hodnote bohuzel napoveda neni k dispozici." crlf
  crlf)
  (retract ?fakt)
)

(defrule vymaz-vstup-nevim
  (declare (salience 15))
  (nacitej)
  (poradi ?vlastnost $?ostatni)
  (hodnota ?vlastnost $?)
  ?fakt <- (vstup ?vlastnost nevim)
  =>
  (retract ?fakt)
)

(defrule vymaz-hodnoty-moznosti
  (declare (salience 12))
  (nacitej)
  (hodnota $?)
  (poradi ? $?poradi)
  ?fakt <- (moznosti ?vlastnost ? $?)
  (test (member$ ?vlastnost $?poradi))
  =>
  (retract ?fakt)
  (assert (moznosti ?vlastnost))
)

(defrule zpracuj-zadanou-vlastnost

```

```

(declare (salience 10))
?faze <- (nacitej)
?fakt <- (poradi ?vlastnost $?ostatni)
?fakt2 <- (hodnota ?vlastnost $?hodnoty)
=>
(retract ?faze ?fakt ?fakt2)
(assert (pokusne-poradi $?ostatni))
(assert (vstup ?vlastnost $?hodnoty))
(assert (priprav))
)

(defrule vse-zadano
  (declare (salience 100))
  ?faze <- (priprav)
  ?fakt <- (pokusne-poradi)
  =>
  (retract ?faze ?fakt)
  (assert (hledej))
)

; ----- Vyjimky -----

; je-li osoba plesata, barva vlasu neni znama
(defrule auto-doplň-vlasy_b
  (declare (salience 5))
  (nacitej)
  (vstup vlasy ples)
  ?fakt <- (poradi $?pred vlasy_b $?za)
  =>
  (retract ?fakt)
  (assert (poradi $?pred $?za))
)

; ----- Vyhledani vysledku -----

(defrule nalezeno
  (declare (salience 10))
  (hledej)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa ?vrasa)
  (vstup vlasy ?vvlasy)
  (vstup vlasy_b ?vvlasy_b)
  (vstup oci ?voci)
  (vstup vousy ?vvousy)
  (vstup dalsi $?vdalsi)
  (tvar (jmeno ?jmeno) (prijmeni ?prijmeni)
    (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
    ))
    (rasa ?rasa&:( or (eq ?rasa ?vrasa ) (eq ?vrasa nevim)
    ))
    (vlasy ?vlasy&:( or (eq ?vlasy ?vvlasy ) (eq ?vvlasy nevim)
    ))
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
    ))
    (oci ?oci&:( or (eq ?oci ?voci ) (eq ?voci nevim)
    ))
    (vousy ?vousy&:( or (eq ?vousy ?vvousy ) (eq ?vvousy nevim)
    ))
  )
)

```



```

        (dalsi  $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
$?vdalsi)))
    )
    =>
    (printout t "Hledana osoba by mohla byt: " ?jmeno " " ?prijmeni "." crlf)
)

(defrule nenalezeno
  (declare (salience 10))
  (hledej)
  (vstup pohlavi ?vpohlavi)
  (vstup rasa    ?vrasa)
  (vstup vlasy   ?vvlasy)
  (vstup vlasy_b ?vvlasy_b)
  (vstup oci     ?voci)
  (vstup vousy   ?vvousy)
  (vstup dalsi   $?vdalsi)
  (not(tvar (jmeno ?jmeno) (prijmeni ?prijmeni)
    (pohlavi ?pohlavi&:(or (eq ?pohlavi ?vpohlavi) (eq ?vpohlavi nevim)
))
    (rasa    ?rasa&:( or (eq ?rasa    ?vrasa    ) (eq ?vrasa    nevim)
))
    (vlasy   ?vlasy&:( or (eq ?vlasy   ?vvlasy   ) (eq ?vvlasy   nevim)
))
    (vlasy_b ?vlasy_b&:(or (eq ?vlasy_b ?vvlasy_b) (eq ?vvlasy_b nevim)
))
    (oci     ?oci&:( or (eq ?oci     ?voci     ) (eq ?voci     nevim)
))
    (vousy   ?vousy&:( or (eq ?vousy   ?vvousy   ) (eq ?vvousy   nevim)
))
    (dalsi   $?dalsi&:( or (member$ $?vdalsi $?dalsi) (member$ nevim
$?vdalsi)))
  ))
  =>
  (printout t "Zadanemu popisu neodpovida zadna z evidovanych osob." crlf)
)

(defrule vypsano-vse
  ?faze <- (hledej)
  =>
  (retract ?faze)
  (assert (uklid))
)

; ----- Odstraneni ci obnoveni pomocnych faktu -----

(defrule obnov-vstupy
  (uklid)
  ?fakt <- (vstup ?vlastnost ?x&~nevim $?)
  =>
  (retract ?fakt)
  (assert (vstup ?vlastnost nevim))
)

(defrule obnov-moznosti
  (uklid)
  ?fakt <- (moznosti ?vlastnost ? $?)
  =>
  (retract ?fakt)
  (assert (moznosti ?vlastnost))
)

```

```

(defrule smaz-poradi
  (uklid)
  ?fakt <- (poradi $?)
  =>
  (retract ?fakt)
)

(defrule obnov-poradi
  (uklid)
  (zaloha-poradi $?poradi)
  =>
  (assert (pokusne-poradi $?poradi))
)

(defrule ukonci
  (declare (salience -10))
  ?faze <- (uklid)
  =>
  (retract ?faze)
  (assert (konec))
)

; ----- Dotaz na pokracovani -----

(defrule ma_se_pokracovat
  ?fakt <- (konec)
  =>
  (bind ?y 2)
  (while (or (not(integerp ?y)) (> ?y 1) (< ?y 0)) do
    (printout t crlf "Prejete si spustit identifikacni program znovu od
zacatku (1 = ano, 0 = ne - konec)?" crlf)
    (bind ?y(read))
  )
  (retract ?fakt)
  (if (= ?y 1) then
    (assert (start))
    (printout t crlf)
  else
    (printout t crlf "KONEC" crlf crlf)
  )
)

```

## Data (data.clp)

```
(tvar (prijmeni alda) (jmeno alan) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni allen) (jmeno woody) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b seda) (oci hneda) (vousy zadne) (dalsi bryle))
(tvar (prijmeni applegate) (jmeno christina) (pohlavi zena) (rasa beloch) (vlasy pod_usi) (vlasy_b blond) (oci modra) (vousy zadne) (dalsi zadne))
(tvar (prijmeni basikova) (jmeno bara) (pohlavi zena) (rasa beloch) (vlasy pod_usi) (vlasy_b zrzava) (oci modra) (vousy zadne) (dalsi zadne))
(tvar (prijmeni bila) (jmeno lucie) (pohlavi zena) (rasa beloch) (vlasy po_ramena) (vlasy_b cerna) (oci hneda) (vousy zadne) (dalsi zadne))
(tvar (prijmeni collins) (jmeno joan) (pohlavi zena) (rasa beloch) (vlasy kstice) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni connery) (jmeno sean) (pohlavi muz) (rasa beloch) (vlasy ples) (vlasy_b nevim) (oci hneda) (vousy vsude) (dalsi kotlety))
(tvar (prijmeni cosby) (jmeno bill) (pohlavi muz) (rasa cernoch) (vlasy kudrlinky) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni falk) (jmeno peter) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b prosedivela) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni fiser) (jmeno bohumil) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b hneda) (oci hneda) (vousy zadne) (dalsi bryle))
(tvar (prijmeni flair) (jmeno ric) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b blond) (oci modra) (vousy zadne) (dalsi zadne))
(tvar (prijmeni ford) (jmeno harrison) (pohlavi muz) (rasa beloch) (vlasy dlouhy_jezek) (vlasy_b prosedivela) (oci modra) (vousy zadne) (dalsi zadne))
(tvar (prijmeni frakes) (jmeno jonathan) (pohlavi muz) (rasa beloch) (vlasy dlouhy_jezek) (vlasy_b cerna) (oci modra) (vousy vsude) (dalsi kotlety))
(tvar (prijmeni freeman) (jmeno morgan) (pohlavi muz) (rasa cernoch) (vlasy dlouhy_jezek) (vlasy_b prosedivela) (oci cerna) (vousy okolo) (dalsi zadne))
(tvar (prijmeni funes) (jmeno louis_de) (pohlavi muz) (rasa beloch) (vlasy ples) (vlasy_b nevim) (oci cerna) (vousy zadne) (dalsi kotlety))
(tvar (prijmeni goldberg) (jmeno whoopi) (pohlavi zena) (rasa cernoch) (vlasy po_ramena) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni gott) (jmeno karel) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi kotlety))
(tvar (prijmeni hogan) (jmeno hulk) (pohlavi muz) (rasa beloch) (vlasy okolo_hlavy) (vlasy_b blond) (oci modra) (vousy knir) (dalsi kotlety))
(tvar (prijmeni houston) (jmeno whitney) (pohlavi zena) (rasa cernoch) (vlasy pod_usi) (vlasy_b hneda) (oci hneda) (vousy zadne) (dalsi zadne))
(tvar (prijmeni chan) (jmeno jackie) (pohlavi muz) (rasa asiat) (vlasy pesinka) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni klaus) (jmeno vaclav) (pohlavi muz) (rasa beloch) (vlasy jezek) (vlasy_b seda) (oci cerna) (vousy knir) (dalsi bryle))
(tvar (prijmeni koranteng) (jmeno rey) (pohlavi muz) (rasa cernoch) (vlasy kudrlinky) (vlasy_b cerna) (oci cerna) (vousy zadne) (dalsi kotlety))
(tvar (prijmeni lamas) (jmeno lorenzo) (pohlavi muz) (rasa beloch) (vlasy po_ramena) (vlasy_b cerna) (oci hneda) (vousy zadne) (dalsi kotlety))
(tvar (prijmeni li) (jmeno jet) (pohlavi muz) (rasa asiat) (vlasy dlouhy_jezek) (vlasy_b cerna) (oci hneda) (vousy zadne) (dalsi zadne))
(tvar (prijmeni novotny) (jmeno petr) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b cerna) (oci hneda) (vousy okolo) (dalsi zadne))
(tvar (prijmeni oneill) (jmeno ed) (pohlavi muz) (rasa beloch) (vlasy dlouhy_jezek) (vlasy_b cerna) (oci hneda) (vousy zadne) (dalsi zadne))
(tvar (prijmeni ruml) (jmeno jan) (pohlavi muz) (rasa beloch) (vlasy ples) (vlasy_b nevim) (oci modra) (vousy vsude) (dalsi kotlety))
(tvar (prijmeni sagal) (jmeno katey) (pohlavi zena) (rasa beloch) (vlasy po_ramena) (vlasy_b zrzava) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni savalas) (jmeno telly) (pohlavi muz) (rasa beloch) (vlasy ples) (vlasy_b nevim) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni schwarzenegger) (jmeno arnold) (pohlavi muz) (rasa beloch) (vlasy dlouhy_jezek) (vlasy_b cerna) (oci modra) (vousy zadne) (dalsi ))
(tvar (prijmeni smith) (jmeno will) (pohlavi muz) (rasa cernoch) (vlasy jezek) (vlasy_b cerna) (oci cerna) (vousy okolo) (dalsi zadne))
(tvar (prijmeni souckova) (jmeno marie) (pohlavi zena) (rasa beloch) (vlasy kstice) (vlasy_b zrzava) (oci modra) (vousy zadne) (dalsi kotlety))
(tvar (prijmeni spidla) (jmeno vladimir) (pohlavi muz) (rasa beloch) (vlasy pesinka) (vlasy_b seda) (oci cerna) (vousy zadne) (dalsi bryle))
(tvar (prijmeni stewart) (jmeno patrick) (pohlavi muz) (rasa beloch) (vlasy ples) (vlasy_b nevim) (oci modra) (vousy zadne) (dalsi zadne))
(tvar (prijmeni "t") (jmeno booker) (pohlavi muz) (rasa cernoch) (vlasy jezek) (vlasy_b cerna) (oci cerna) (vousy knir) (dalsi zadne))
(tvar (prijmeni zazvorkova) (jmeno stella) (pohlavi zena) (rasa beloch) (vlasy po_ramena) (vlasy_b seda) (oci cerna) (vousy zadne) (dalsi zadne))
(tvar (prijmeni zeman) (jmeno milos) (pohlavi muz) (rasa beloch) (vlasy dlouhy_jezek) (vlasy_b seda) (oci modra) (vousy zadne) (dalsi zadne))
```

## Help (help.clp)

```
(help pohlavi      "Je treba zadat, jakeho pohlavi je hledana osoba.")
(help pohlavi muz  "Hledana osoba je muz.")
(help pohlavi zena "Hledana osoba je zena.")
(help rasa        "Je treba zadat, jake rasy hledana osoba je.")
(help rasa beloch "Hledana osoba je beloch - ma bilou plet.")
(help rasa cernoch "Hledana osoba je cernoch - ma cernou plet.")
(help rasa asiat  "Hledana osoba je asiat.")
(help vlasy       "Je treba zadat, jak ma hledana osoba upravene vlasy.")
(help vlasy ples  "Hledana osoba nema zadne vlasy, ci jen velmi kratoucke a proridle na temeni hlavy.")
(help vlasy jezek "Hledana osoba ma vlasy ostrihane na jezka - tedy velmi kratke (0,5 - 2 cm).")
(help vlasy dlouhy_jezek "Hledana osoba ma vlasy ostrihane na delsiho jezka - tedy kratke (2 - 4 cm).")
(help vlasy po_ramena "Hledana osoba ma vlasy dlouhe az po ramena.")
(help vlasy pesinka "Hledana osoba ma vlasy dlouhe prave natolik, aby si je mohla ucesat do pesinky (4 - 7 cm).")
(help vlasy kstice "Hledana osoba ma delsi vlasy smerujici vzhuru (kudrlinky, mismas apod).")
(help vlasy pod_usi "Hledana osoba ma delsi vlasy, ktere ji dosahuji az pod usi.")
(help vlasy okolo_hlavy "Hledana osoba ma delsi vlasy ovsem pouze okolo hlavy a na jejim vrsku je plesata.")
(help vlasy_b      "Je treba zadat, jakou barvu vlasu hledana osoba ma.")
(help oci         "Je treba zadat, jakou barvu oci hledana osoba ma.")
(help vousy       "Je treba zadat, ma-li hledana osoba nejake vousy a pokud ano, tak jak je ma upravene.")
(help vousy zadne  "Hledana osoba nema zadne vousy.")
(help vousy knir   "Hledana osoba ma knir (vousy pouze pod nosem).")
(help vousy okolo  "Hledana osoba ma vousy okolo ust.")
(help vousy vsude  "Hledana osoba ma vousy vsude, tedy nejen okolo ust, ale sahaji ji az k vlasum.")
(help dalsi       "Je treba zadat, jake dalsi vzhledove vlastnosi ma tvar hledane osoby.")
(help dalsi zadne  "Tvar hledane osoby nema zadne dalsi z nabizenych vzhledovych vlastnosti.")
(help dalsi bryle  "Hledana osoba nosi bryle.")
(help dalsi kotlety "Hledana osoba ma kotlety - kratoucke vlasy vyrustajici pred usima na tvari.")
```